REMARKS:

Restriction Requirement

Restriction to one of the following groups was required: Group I including claims 1 to 53, 80 to 110, 124 and 125, and Group II including claims 54 to 79 and 11 to 123. As noted in the Office Action, Applicants elected Group I via telephone. Applicants affirm this election. Furthermore, Applicants have cancelled the non-elected claims in Group II.

The Office Action indicated that elected Group I was drawn to "instruction analyzing and verification" and that non-elected Group II was drawn to "network accessing and regulation." Applicants have added new claims 126 to 164 that correspond to the claims from Group II amended to be specifically drawn to instruction analyzing and verification. Thus, examination of these new claims with the claims in Group I is believed to be proper.

Status

Claims 1 to 53, 80 to 110, and 124 to 164 are pending. Claims 54 to 79 and 111 to 123 have been cancelled. Claims 124 to 164 have been added. Claims 1, 18, 36, 80, 90, 100, 110, 124, 125, 126, 134, 142, 152, 153, and 157 are the independent claims. Reconsideration and further examination are respectfully requested.

Section 102 Rejections

Claims 1 to 53, 80 to 110, 124 and 125 were rejected under 35 USC § 102 over U.S. Patent No. 6,151,618 (Wahbe). Applicants have amended the independent ones of these claims. The rejected claims are discussed below grouped by independent claims.

Claims 1 to 17: Independent claim 1 is reproduced below as amended:

> 1. A method of analyzing instructions and data for a program to determine where the instructions and data might result in incorrect results when run on a multiprocessor system, the method comprising the steps of:
> dividing the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data, the multiprocessor system configured to use at most one processor at a time to execute instructions and to access data from any one domain;
> determining which of the instructions and data involve references outside of their domains;
> determining which of the references outside of their domains are multiprocessor unsafe references;
> generating a report of the multiprocessor unsafe references; and
> modifying the instructions and data based on the report.

The applied art does not show the foregoing features of claim 1, at least with respect to "dividing the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data."

In more detail, Wahbe deals with a "safe general purpose virtual machine computing system" (Abstract) that ensures a program running under the system does not perform unsafe operations, for example accessing memory outside memory defined by the program's memory access permission. The Office Action apparently equated this memory with the claimed domain. However, ensuring that a program stays within memory defined for the program is entirely different from dividing a program into plural domains for multiprocessing. Wahbee does not discuss having one

program divided into plural domains for multiprocessor execution and then determining if the program will run properly in those plural domains.

Claim 1 recites such operations, namely "dividing the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data" and then performing other steps (i.e., determining, generating and modifying) to ensure that the program will run properly in those plural domains. Thus, Wahbe does not teach the invention as recited by amended claim 1.

For at least the foregoing reasons, claim 1 and its dependent claims are believed to be allowable over Wahbe. Such action is respectfully requested.

Applicants note that claim 4 is also clearly different from Wahbe. This claim recites that "the instructions and data comprise source code that is analyzed before compilation." The Office Action points to Fig. 7, step 710, and Fig. 4, step 434 of Wahbee as teaching this feature. However, step 710 in Wahbee is "CONVERT SOURCE LANGUAGE PROGRAM TO VM PROGRAM INSTRUCTIONS." Here, "VM" stands for "Virtual Machine." This conversion occurs before Wahbee's step 718, "IDENTIFY USAGE INSTRUCTIONS AMONG THE VM PROGRAM INSTRUCTIONS." Thus, in Wahbee, the coverted VM program instructions are checked, not the source code. Applicant's contention in this regard is supported by Wahbee's statement at col. 13, lines 36 and 37, that "[a]t step 718, unsafe instructions are identified among the virtual machine instructions generated by step 710."

Allowance of claim 4 is believed to be proper for this additional reason as well. Such action is respectfully requested.

Claims 18 to 35: Independent claim 18 is reproduced below as amended:

> 18. A memory storing information including steps executable by a processor, the steps executable to analyze instructions and data for a program to determine where the instructions and data might result in incorrect results when run on a multiprocessor system, the steps comprising:
> dividing the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data, the multiprocessor system configured to use at most one processor at a time to execute instructions and to access data from any one domain;
> determining which of the instructions and data involve references outside of their domains;
> determining which of the references outside of their domains are multiprocessor unsafe references;
> generating a report of the multiprocessor unsafe references; and
> modifying the instructions and data based on the report.

Substantially as discussed above with respect to claim 1, the applied art does not show the foregoing features of claim 18, at least with respect to "dividing the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data." Furthermore, substantially as discussed above with respect to claim 4, Wahbe does not teach claim 21's feature that "the instructions and data comprise source code that is analyzed before compilation." Accordingly, allowance of claim 18 and its dependent claims over Wahbe is respectfully requested.

Claims 36 to 53: Independent claim 36 is reproduced below as amended:

> 36. An analyzer that analyzes instructions and data for a program to determine where the instructions and data might result in incorrect results when run on a multiprocessor system, the analyzer comprising:
> a reference analyzer that divides the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data, the multiprocessor system configured to use at most one processor at a time to execute instructions and to access data from any

one domain, that determines which of the instructions and data involve references outside of their domains, and that determines which of the references outside of their domains are multiprocessor unsafe references; and

a report generator that generates a report of the multiprocessor unsafe references.

Substantially as discussed above with respect to claim 1, the applied art does not show the foregoing features of claim 36, at least with respect to "a reference analyzer that divides the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data." Furthermore, substantially as discussed above with respect to claim 4, Wahbe does not teach claim 40's feature that "the instructions and data comprise source code that is analyzed before compilation." Accordingly, allowance of claim 36 and its dependent claims over Wahbe is respectfully requested.

Claims 80 to 89: Independent claim 80 is reproduced below as amended:

80. A method of analyzing instructions and data and dynamically determining where the instructions and data for a program might result in incorrect results when run on a multiprocessor system, the method comprising the steps of:

dividing the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data, the multiprocessor system configured to use at most one processor at a time to execute instructions and to access data from any one domain;

determining which of the instructions and data involve references outside of their domains;

determining which of the references outside of their domains are purportedly multiprocessor safe references;

generating a table of the purportedly multiprocessor safe references, the table including the domains to which the references are supposed to refer;

executing the instructions and data; and

when a reference in the table of purportedly microprocessor safe references is encountered during execution of the instructions and data,

determining if the reference is actually to a domain to which that reference is supposed to refer.

Substantially as discussed above with respect to claim 1, the applied art does not show the foregoing features of claim 80, at least with respect to "dividing the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data." Accordingly, allowance of claim 80 and its dependent claims over Wahbe is respectfully requested.

Claims 90 to 99: Independent claim 90 is reproduced below as amended:

90. A memory storing information including steps executable by a processor, the steps executable to analyze instructions and data for a program and dynamically determine where the instructions and data might result in incorrect results when run on a multiprocessor system, the steps comprising:

dividing the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data, the multiprocessor system configured to use at most one processor at a time to execute instructions and to access data from any one domain;

determining which of the instructions and data involve references outside of their domains;

determining which of the references outside of their domains are purportedly multiprocessor safe references;

generating a table of the purportedly multiprocessor safe references, the table including the domains to which the references are supposed to refer;

executing the instructions and data; and

when a reference in the table of purportedly microprocessor safe references is encountered during execution of the instructions and data, determining if the reference is actually to a domain to which that reference is supposed to refer.

Substantially as discussed above with respect to claim 1, the applied art does not show the foregoing features of claim 90, at least with respect to "dividing the instructions and data for the

program into plural domains based on symbols used to refer to those instructions and data." Accordingly, allowance of claim 90 and its dependent claims over Wahbe is respectfully requested.

Claims 100 to 109: Independent claim 100 is reproduced below as amended:

> 100. A system for analyzing instructions and data for a program and dynamically determining where the instructions and data might result in incorrect results when run on a multiprocessor system, the system comprising:
>
> a reference analyzer that divides the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data, the multiprocessor system configured to use at most one processor at a time to execute instructions and to access data from any one domain, that determines which of the instructions and data involve references outside of their domains, and that determines which of the references outside of their domains are purportedly multiprocessor safe references;
>
> a table generator that generates a table of the purportedly multiprocessor safe references, the table including the domains to which the references are supposed to refer;a reference tracker that tracks references made by the instructions and data; and
>
> a comparator that determines, when a reference in the table of purportedly microprocessor safe references is encountered during execution of the instructions and data, if the reference is actually to a domain to which that reference is supposed to refer.

Substantially as discussed above with respect to claim 1, the applied art does not show the foregoing features of claim 100, at least with respect to "a reference analyzer that divides the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data." Accordingly, allowance of claim 100 and its dependent claims over Wahbe is respectfully requested.

Claim 110: Independent claim 110 is reproduced below as amended:

> 110. A method of analyzing instructions and data for a program to determine where the instructions and data might result in incorrect results

when run on a system having multiple resources of a type used concurrently, the method comprising the steps of:

dividing the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data, the system configured to use at most one of the resources at a time to execute instructions and to access data from any one domain;

determining which of the instructions and data involve references outside of their domains;

determining which of the references outside of their domains are unsafe references;

generating a report of the unsafe references; and

modifying the instructions and data based on the report.

Substantially as discussed above with respect to claim 1, the applied art does not show the foregoing features of claim 110, at least with respect to "dividing the instructions and data for the program into plural domains based on symbols used to refer to those instructions and data." Accordingly, allowance of claim 110 and its dependent claims over Wahbe is respectfully requested.

Claim 124: Independent claim 124 is reproduced below as amended:

124. A report stored in a memory, the report resulting from analysis of instructions and data for a program to determine where the instructions and data might result in incorrect results when run on a multiprocessor system, the report comprising:

a division of the instructions and data for the program into plural domains based on the symbols used to refer to those instructions and data, the multiprocessor system configured to use at most one processor at a time to execute instructions and to access data from any one domain;

a list of inter-domain references by the instructions and data that the analysis has shown are multiprocessor unsafe; and

for each inter-domain reference, the domains involved in the inter-domain reference.

Substantially as discussed above with respect to claim 1, the applied art does not show the foregoing features of claim 124, at least with respect to "a division of the instructions and data for the program into plural domains based on the symbols used to refer to those instructions and

data." Accordingly, allowance of claim 124 and its dependent claims over Wahbe is respectfully requested.

Claim 125: Independent claim 125 is reproduced below as amended:

> 125. A table stored in a memory, the table resulting from analysis of instructions and data for a program to determine where the instructions and data might result in incorrect results when run on a multiprocessor system, the report comprising:
> a division of the instructions and data for the program into plural domains based on the symbols used to refer to those instructions and data, the multiprocessor system configured to use at most one processor at a time to execute instructions and to access data from any one domain;
> a list of purportedly microprocessor safe references by the instructions and data outside of their domains; and
> the domains to which the references are supposed to refer.

Substantially as discussed above with respect to claim 1, the applied art does not show the foregoing features of claim 125, at least with respect to "a division of the instructions and data for the program into plural domains based on the symbols used to refer to those instructions and data." Accordingly, allowance of claim 125 and its dependent claims over Wahbe is respectfully requested.

New Claims

The language of the new claims is intended to distinguish those claims from Wahbe. In particular, the independent ones of the new claims recite features related to dividing a program into plural processing domains. These claims therefore also are believed to be allowable over Wahbe, and such action is respectfully requested.

No Admission

Applicants' decision not to argue each of the dependent claims separately is not an admission that the subject matter of those claims is taught by the applied art.

Closing

In view of the foregoing amendments and remarks, the entire application is believed to be in condition for allowance, and such action is respectfully requested at the Examiner's earliest convenience.

Applicants' undersigned attorney can be reached at (614) 205-3241. All correspondence should continue to be directed to the address indicated below.

Respectfully submitted,

Dane C. Butzer

Dated: August 11, 2005

Dane C. Butzer
Reg. No. 43,521

Swernofsky Law Group PC
P.O. Box 390013
Mountain View, CA 94039-0013
(650) 947-0700